

# In volo sopra l'aeroporto "Caproni" (parte 2)

## Tutorial per Scratch 2.0



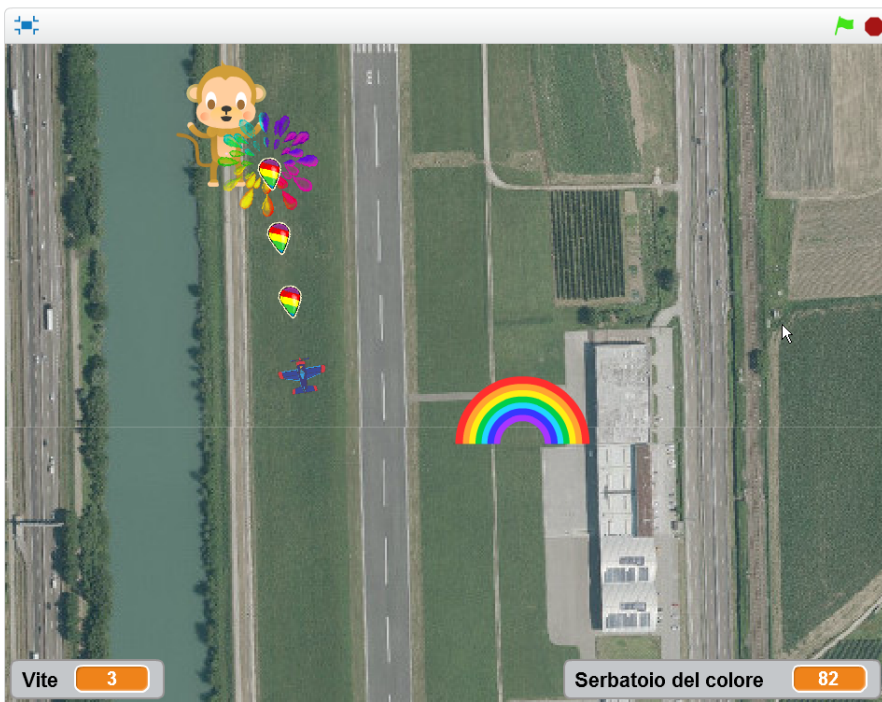
### INTRODUZIONE

Nella prima parte del tutorial abbiamo preparato il campo di volo per il nostro aereo: trovi il progetto da cui partire ([tutorial-14-festivolare-parte-1.sb2](#)) sul nostro sito [coderdojotrento.it/scratch14](http://coderdojotrento.it/scratch14). Ora che possiamo girare sopra l'aeroporto "Caproni" di Mattarello, poco a sud di Trento, introduciamo dei "nemici" ma anche dei personaggi che staranno dalla nostra parte.

In questo tutorial infatti capiremo come aggiungere al gioco degli scimmioni un po' agitati che tenteranno di far cadere il nostro aereo; per dar loro una calmata utilizzeremo una paintball che potrà essere ricaricata passando sopra alcuni arcobaleni.

Come accade in ogni gioco che si rispetti, avremo un numero di vite a disposizione ed un serbatoio del colore non infinito.

Sul sito di CoderDojo Trento trovi la versione finale del gioco, ma se ci metti un po' di fantasia il tuo gioco sarà ancora migliore!



### PREREQUISITI

- \* Scratch 2.0 (installato sul tuo computer oppure utilizzato direttamente [sul sito web](#))
- \* Scarica dal nostro sito [coderdojotrento.it/scratch14](http://coderdojotrento.it/scratch14) il file ZIP "tutorial-14-festivolare.zip", ed estrai i contenuti in una cartella del tuo computer
- \* Carica poi in Scratch il progetto di partenza "tutorial-14-festivolare-parte-1.sb2"

### PASSI

Nel corso del tutorial seguiremo questi passaggi:

1. Preparazione progetto
2. Mettiamo il pilota automatico ed aggiungiamo il rombo dell'aereo
3. Introduciamo un super-cattivo!
4. Un cattivo che si rispetti...
5. Rispondiamo con un po' di colore
6. Se colpiamo King Kong
7. Tante gocce, tanti King Kong...
8. Gestiamo più vite per l'aeroplano
9. Risorse limitate
10. Migliorie varie e challenge



## 1. Preparazione progetto

Carichiamo in Scratch il progetto “Festivolare - parte 1.sb2” costruito grazie alla prima parte del tutorial (se usi Scratch direttamente dal sito, dovrai importare il progetto): salva poi il progetto assegnandogli il nuovo nome “Festivolare - parte 2”.

In quella prima versione del gioco, una volta avviato con la bandierina verde, eravamo in grado di muovere l'aereo utilizzando i tasti cursore; inoltre lo sfondo con le immagini dell'aeroporto scorreva sotto di noi. È un buon punto di partenza, ma dobbiamo rendere il gioco molto più ricco!

## 2. Mettiamo il pilota automatico ed aggiungiamo il rombo dell'aereo

Modifichiamo lo script dell'aeroplano come indicato nella figura ①

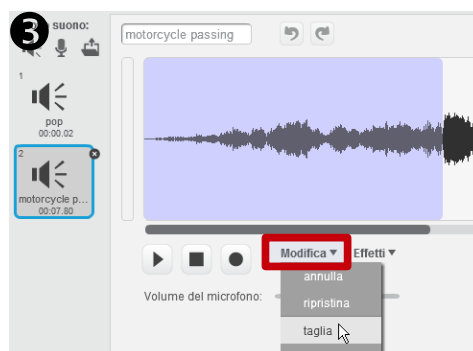
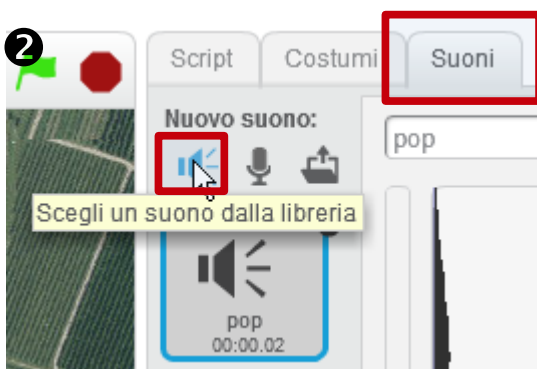
Per fare in modo che l'aereo parta subito, all'inizio del gioco, anche senza premere il tasto “freccia su” sarà sufficiente togliere il blocco “se” che controllava l'istruzione “fai 3 passi”.

Seguendo i passi ②, ③, ④ e ⑤ illustrati qui sotto, possiamo ora aggiungere il rumore del motore:

② dal tab dei suoni, possiamo andare a caricare dalla libreria il suono “motorcycle passing” (che troviamo nella categoria “Effetti”);

③ modifichiamo poi il suono, tagliando l'inizio e la fine per ottenere un bel rombo;

④ per rendere più chiaro il programma, ricordiamo anche di dare sempre dei nomi significativi alle variabili ed agli oggetti: in questo caso cambieremo il nome “motorcycle passing” in “motore”.



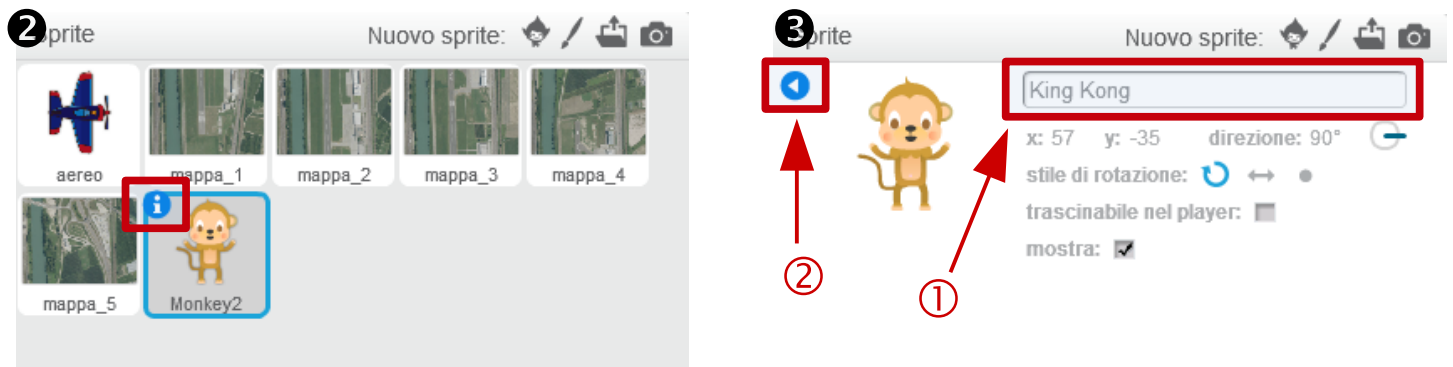
⑤ Una volta costruito il nostro suono, lo riprodurremo in un ciclo, non appena il nostro aereo comincerà a muoversi.



### 3. Introduciamo un super-cattivo!

Non può che essere King Kong: gli aerei infatti, gli davano particolarmente fastidio...

Troviamo qualcosa di mooolto simile a King Kong nella libreria di personaggi di Scratch: utilizzando l'icona del menù indicata nell'immagine ❶ aggiungiamo "Monkey2" (lo sprite è nella categoria "Animali") e per prima cosa apriamo la sezione delle informazioni sullo sprite (vedi immagine ❷) e cambiamo il nome in "King Kong" (immagine ❸).



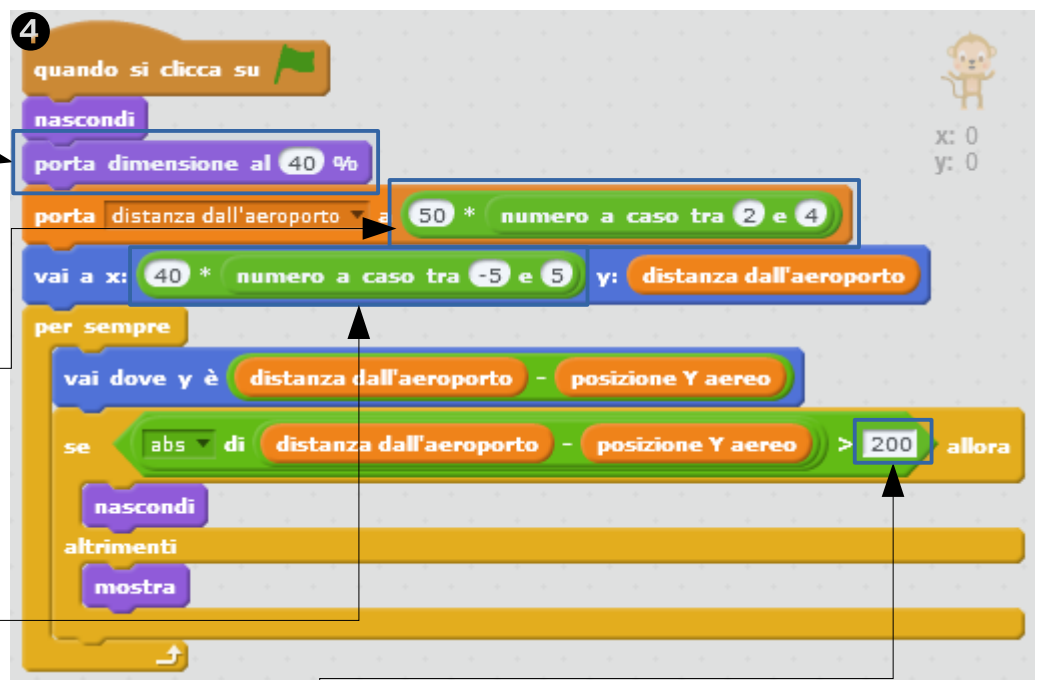
Anche King Kong avrà la sua posizione, quindi in maniera simile a quanto fatto ad esempio per mappa\_2 definiremo una variabile "distanza dall'aeroporto" che varrà solo per questo sprite. Lo script che controlla lo spostamento rispetto all'aereo è molto simile a quella della mappa\_2, quindi compiamo quello script sullo sprite di King Kong e modifichiamolo come nell'immagine ❹

Queste le variazioni:

\* la dimensione iniziale;


\* il valore di "distanza dall'aeroporto" non è fissato come per le mappe, ma un numero casuale che posiziona King Kong poco sopra il punto di partenza dell'aereo;

\* anche la posizione orizzontale (coordinata X) è casuale;



\* la distanza dal centro dello schermo che utilizziamo per decidere se mostrare o nascondere lo sprite è più piccola di quella usata per la mappa: se King Kong si allontana di 200 pixel, dobbiamo nascondere il suo sprite.

N.B Nella pagina seguente spieghiamo meglio il significato di questo numero.

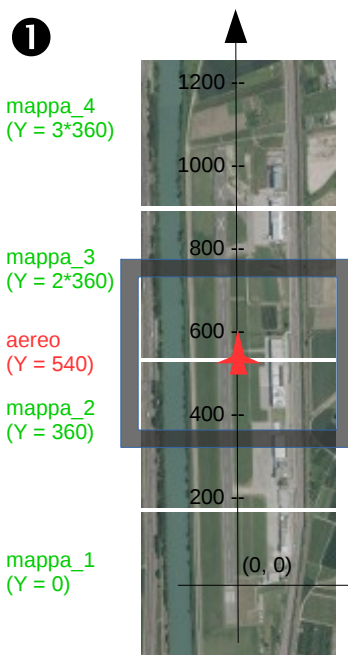
Ricordati sempre che ogni tanto devi salvare il gioco, per evitare di perdere le modifiche in caso di contrattempi (rete che non risponde, computer che fa i capricci, ...). Ricordati anche di provare le modifiche che introduci, man mano che procedi nello sviluppo del gioco: ad esempio se ora lo lanci con , dovresti vedere che -da qualche parte a nord- c'è King Kong che ci aspetta. Ci fa poca paura, è vero, perché non può farci nulla... Rimediamo subito.



## Approfondimento: come funziona lo scroll del fondale?



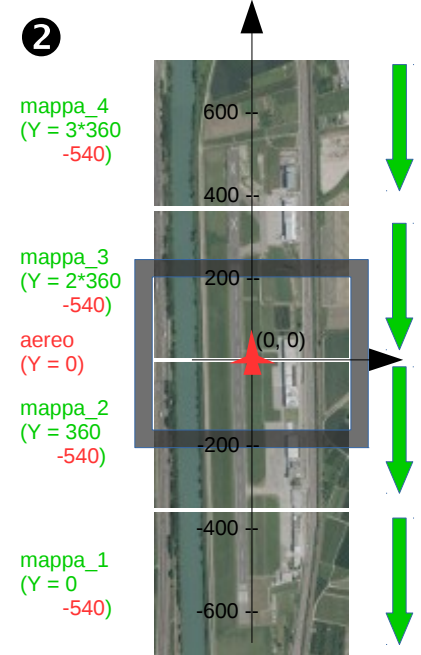
Nella prima parte del tutorial questo argomento è già stato discusso, ma anche senza ripercorrere tutti i passi di quel tutorial, possiamo capire il funzionamento dello scroll che stiamo utilizzando per far muovere il fondale. Posizionando i vari sprite mappa\_1, mappa\_2, mappa\_3, mappa\_4 e mappa\_5 a diverse altezze Y (ovvero diversi valori della variabile privata "distanza dall'aeroporto"), abbiamo costruito una super-mappa fatta di cinque tasselli.



Per ogni mappa e per l'aereo è indicata la coordinata Y, ovvero la loro distanza lungo la verticale dal punto iniziale del decollo, che abbiamo fissato al centro della mappa\_1.

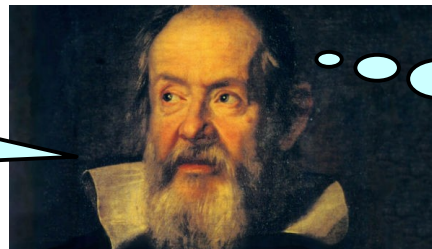
Se potessimo far scorrere il monitor assieme all'aereo che si sposta verso nord, avremo la situazione **1** qui a sinistra (fingiamo che la Y dell'aereo sia).

Ma il monitor non può essere spostato, così -per simulare lo scorrimento verso l'alto dell'aereo- possiamo far scivolare nella direzione opposta tutte le mappe, tenendo fermo l'aereo a Y=0 (vedi figura **2** a destra).



In questo secondo caso, la coordinata di ciascuna mappa è la differenza tra il valore della variabile "distanza dall'aeroporto" e la coordinata Y dell'aereo: questa infatti è la formula della coordinata Y per tutti gli sprite che devono fare lo scroll sotto l'aereo (vedi comando "vai dove Y è" nel ciclo "per sempre" dell'immagine **4** di pagina 3).

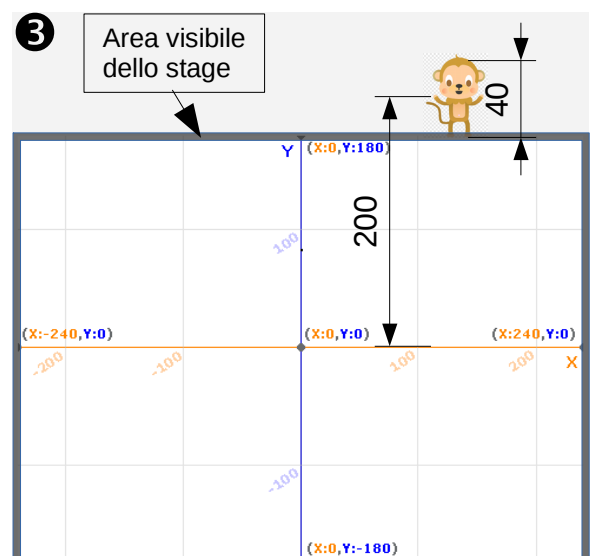
È proprio quello che io chiamo **moto relativo rispetto all'aereo**: se volete ve lo spiego!



... e voi in cambio mi spiegate cosa sia, questo cosiddetto "aereo" ...

Inoltre dobbiamo nascondere gli sprite che finiscono oltre lo schermo: se proviamo a far allontanare uno sprite al di fuori dello stage, Scratch ci impedisce di assegnargli una posizione troppo spostata, in modo che almeno un pezzettino dello sprite sia ancora visibile. Se però vogliamo evitare di vedere le zampe di King Kong penzolare dal bordo superiore dello schermo, o le mappe che si impilano come delle carte da gioco dobbiamo nascondersi.

Nasconderemo quindi uno sprite, quando la sua distanza dal centro dello schermo (che per noi è la differenza tra il valore della variabile "distanza dall'aeroporto" e la coordinata Y dell'aereo) supera  $180 +$  circa metà dell'altezza dello sprite. Vedi **3** e il blocco "se / allora" dell'immagine **4** di pagina 3.



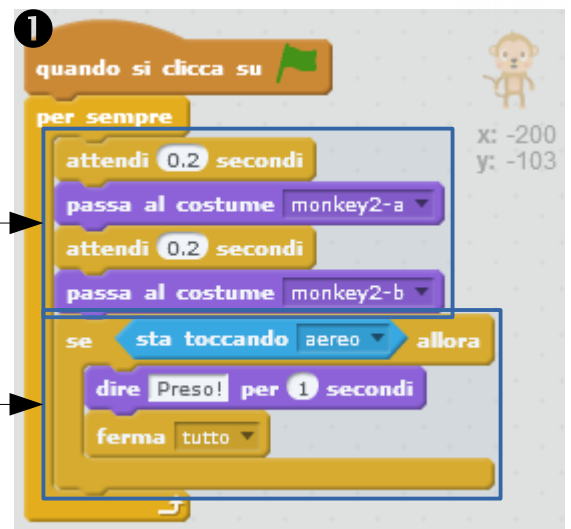
## 4. Un cattivo che si rispetti...

... minimo minimo si agita da matti, e procura dei danni al nostro aereo. Aggiungiamo quindi ai suoi script il blocco di codice descritto nell'immagine ①

\* sfruttiamo due dei suoi tre costumi per farlo sbracciare come un forsennato;

\* usiamo il sensore "sta toccando" per capire se il nostro aereo è a portata di zampa: in quel caso fa finire il gioco, da buon cattivo.

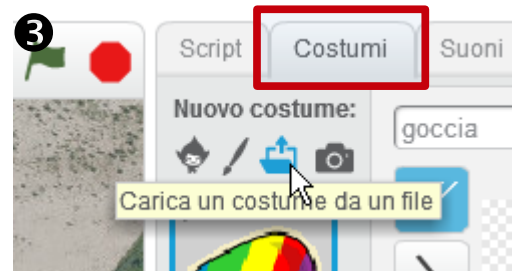
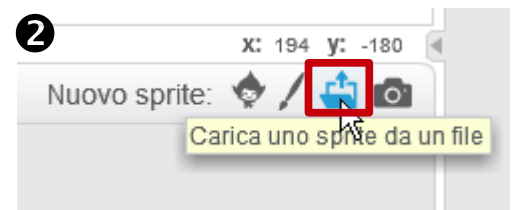
Ma in qualche modo dobbiamo fargli passare l'arrabbiatura: procuriamoci quindi delle gocce di arcobaleno e ridiamogli il buonumore.



## 5. Rispondiamo con un po' di colore

Utilizziamo l'icona del menù indicata nell'immagine ② per creare un nuovo personaggio a partire dal file "goccia.png".

A questo sprite aggiungiamo poi un secondo costume: dal tab "Costumi" (vedi figura ③), usiamo l'icona che permette la creazione di un nuovo costume partendo da un'immagine salvata in un file. Prepariamo quindi il costume "esplosione" caricando il file omonimo.



Vediamo ora lo script della "goccia" (immagine ④). Per utilizzare la goccia come proiettile "a ripetizione" dobbiamo creare diverse copie dello stesso sprite:

\* all'avvio del gioco nascondiamo lo sprite e prepariamo il giusto costume e la dimensione;

\* in un ciclo "per sempre", poi, controlliamo la pressione della barra di spazio;

\* se questa è premuta, produciamo un suono e creiamo una copia identica dello sprite "goccia"; attendiamo poi qualche frazione di secondo per evitare che la raffica sia troppo fitta;

\* ogni clone che viene creato esegue il blocco di codice "quando vengo clonato" (proprio come tutti gli sprite eseguono i blocchi "quando si clicca sulla bandierina verde", all'avvio del gioco): per ora ci limitiamo a posizionare ogni clone dove sta l'aereo e lo rendiamo visibile.



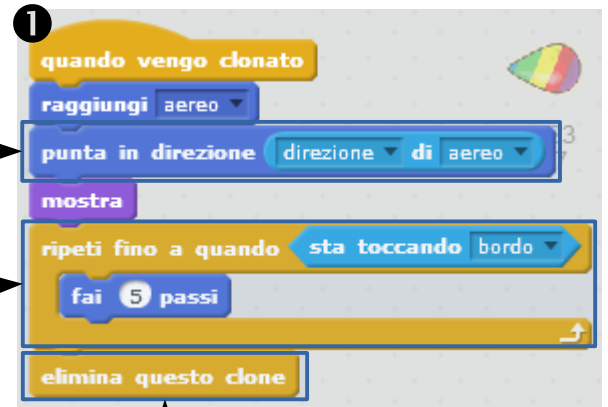
Se adesso provi a lanciare il gioco, muovi l'aereo a destra e sinistra e provi a premere la barra di spazio, vedrai che l'aereo lascia sulla sua strada i cloni dello sprite "goccia" come se depositasse delle uova... :-D Proiettili un po' scarsi, in effetti!

Per aggiungere il movimento ai vari cloni dello sprite "goccia", dobbiamo modificare il suo script come indicato nell'immagine ① qui a destra:

\* prima di mostrare la goccia, la orientiamo proprio nella direzione dell'aereo (trovi il blocco azzurro chiaro nella sezione dei sensori);

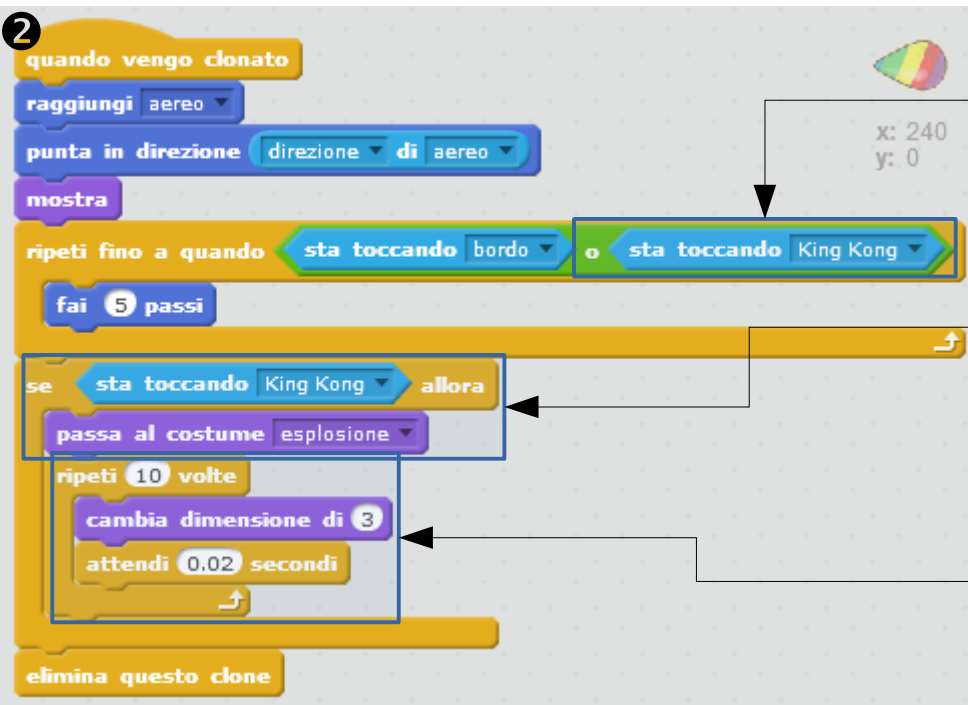
\* in un ciclo "ripeti fino a quando", facciamo avanzare la goccia finché non tocca il bordo dello stage;

\* fuori dal ciclo, ovvero una volta raggiunto il bordo, dobbiamo distruggere il clone; dobbiamo infatti pensare che -ogni volta che spariamo una goccia di colore- viene creata una copia identica dello sprite: non possiamo quindi continuare a creare personaggi, senza poi rilasciare le risorse, quando non ci servono più.



Prova a far partire il gioco e controlla il comportamento dei proiettili-goccia. E salva!

Aggiungiamo adesso la gestione del contatto con King Kong, modificando lo script precedente: lo script finale è quello descritto nell'immagine ②. Queste le modifiche:



\* adesso la goccia si sposta in avanti fino a che il contatto non avviene con il bordo oppure con King Kong;

\* se si è fermata proprio a causa dello scontro con King Kong, cambia il costume mettendo quello dell'esplosione;

\* in un piccolo ciclo, avviene un progressivo aumento della dimensione, in modo che venga simulato l'allargamento dello schizzo.

Vediamo ora come reagisce King Kong quando viene colpito.

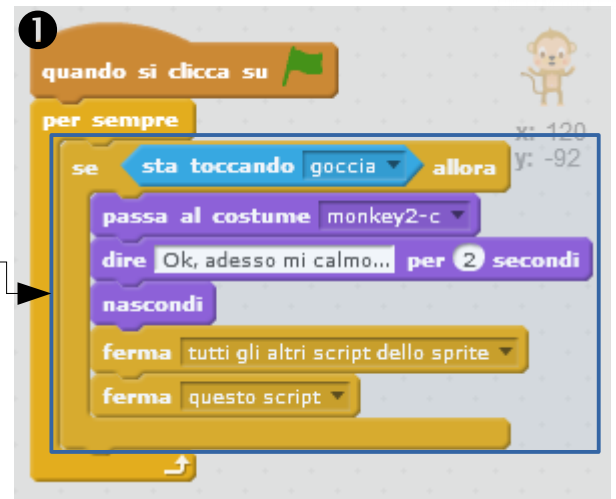
## 6. Se colpiamo King Kong

Vediamo adesso come arricchire gli script di King Kong, quando questo viene colpito da una delle gocce (vedi immagine 1).

Prepariamo un nuovo ciclo “per sempre” che parte con la bandierina verde, e nel quale viene effettuato il seguente controllo:

\* se King Kong “sta toccando” lo sprite “goccia” allora cambia costume, dichiara la sua resa e ferma tutti gli script dello sprite.

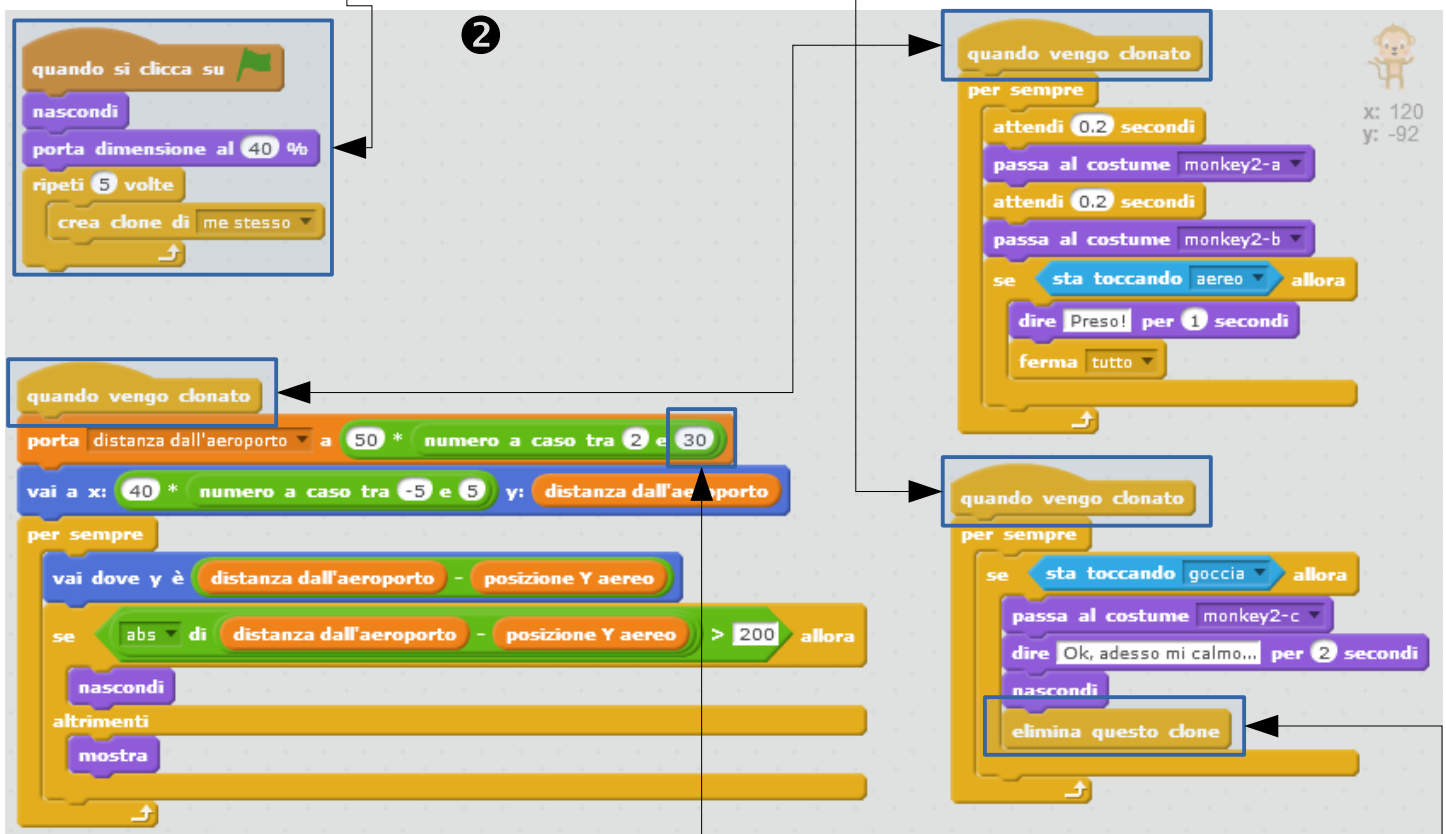
Prova il gioco e verifica che funzioni tutto!



## 7. Tante gocce, tanti King Kong...

Adesso che abbiamo capito come utilizzare i cloni per creare le “gocce a raffica”, perché non utilizzare lo stesso meccanismo per generare un branco intero di gorilla giganti?

Dobbiamo quindi modificare gli script di King Kong nel modo descritto nell'immagine 2: l'unico blocco di codice che parte con la bandierina verde contiene in pratica solo il ciclo che crea i cloni; gli altri blocchi che prima partivano con un “quando si clicca su” ora dovranno girare per ciascun clone, quindi si attiveranno con un “quando vengo clonato”.



Ci sono altre due piccole modifiche:

\* la prima riguarda la coordinata Y alla quale mettiamo i cloni di King Kong: adesso copiamo tutte e cinque le schermate;

\* la seconda riguarda il modo in cui fermiamo lo script del King Kong colpito: è sufficiente eliminare il clone ed il gioco è fatto.

Salva il gioco e provalo!



## 8. Gestiamo più vite per l'aeroplano

Se non vogliamo che il gioco finisca al primo incontro ravvicinato con uno dei King Kong presenti sul campo, dobbiamo per prima cosa introdurre una variabile utilizzabile da tutti gli sprite che tenga appunto traccia del numero di vite che abbiamo a disposizione.

Torniamo quindi sullo sprite dell'aereo e -come indicato nella figura ①- creiamo una variabile che chiameremo "Vite" (e che sia di tipo "Per tutti gli sprite").

Lasciamo la spunta a fianco della variabile, in modo che questa venga visualizzata sullo schermo durante il gioco. Trascina poi l'etichetta nell'angolo in basso a sinistra, in modo che possa essere sempre tenuta sott'occhio (vedi immagine ②).



Lasciamo che sia l'aereo (vedi immagine ③) ad occuparsi del settaggio iniziale della variabile: fissiamo questo valore a 3.

Modifichiamo inoltre il blocco che si occupa del volo dell'aereo: questo potrà muoversi solo finché il numero di vite non sarà sceso a zero.

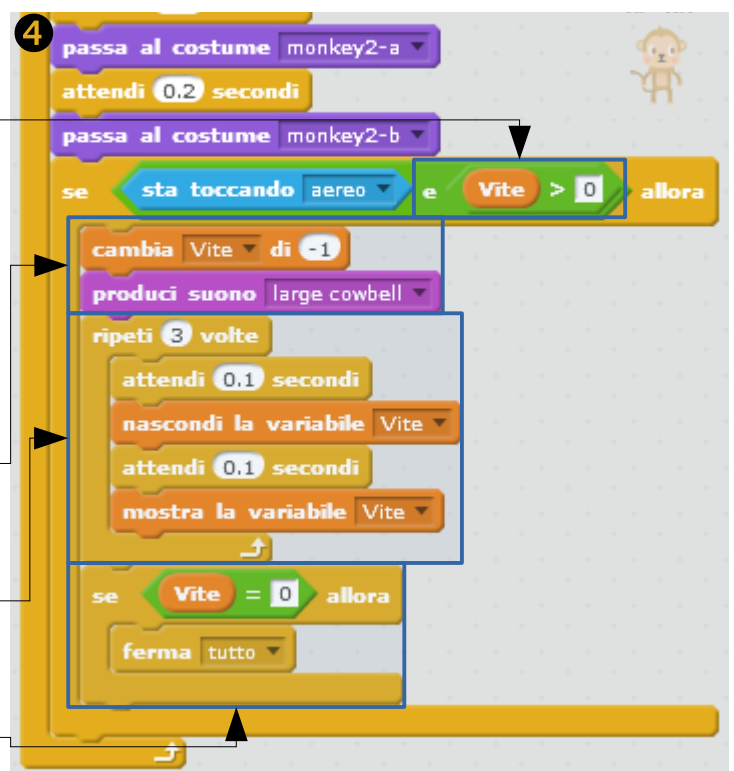
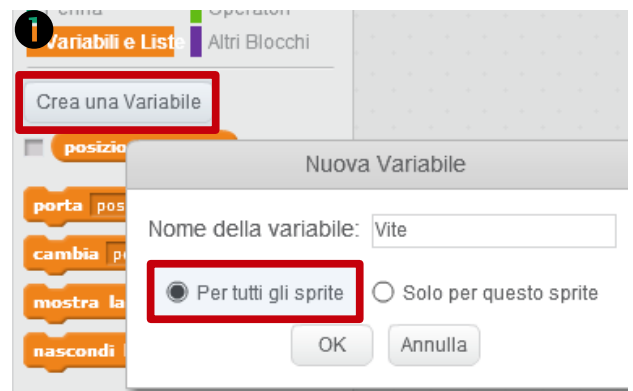
Al contempo, saranno i nostri nemici ad occuparsi di decrementare le vite: andiamo sullo script di King Kong e modifichiamo come indicato nell'immagine ④ il blocco "se" che si occupava del contatto con l'aereo:

\* eseguiamo il blocco di codice del contatto solo se ci sono ancora vite da scalare;

\* decrementiamo il numero di vite ed emettiamo un suono sinistro;

\* facciamo lampeggiare l'etichetta delle vite, così il giocatore sarà allertato della situazione;

\* fermiamo tutto, ma solo se abbiamo davvero esaurito le vite del giocatore.

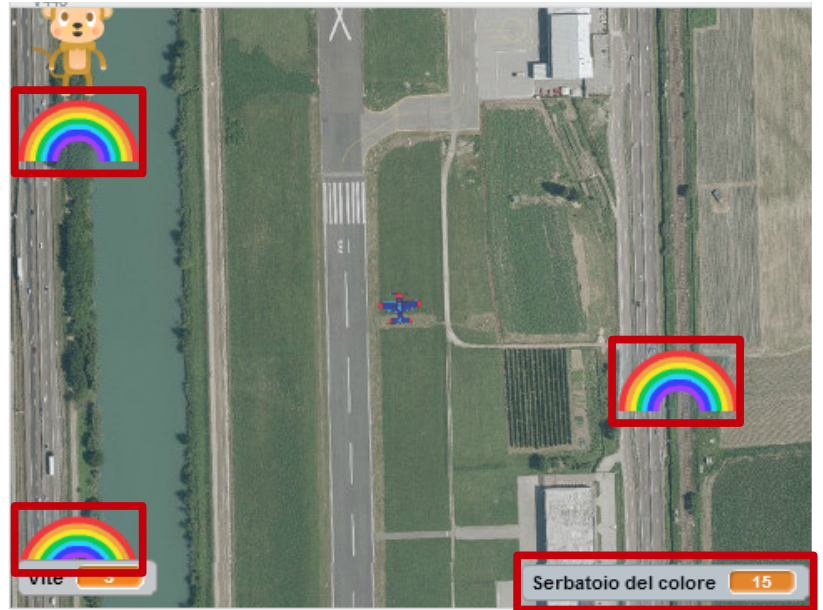




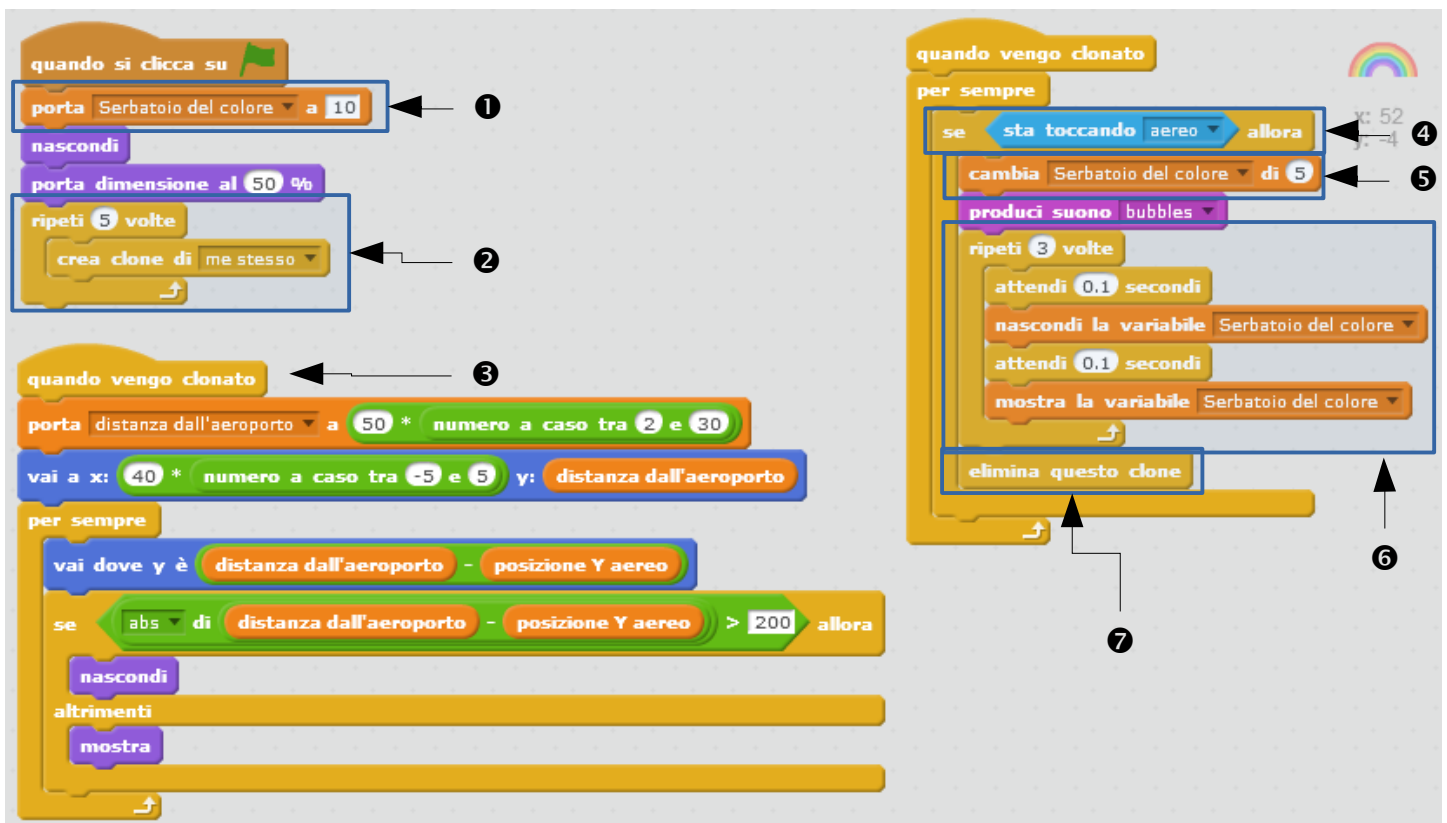
## 9. Risorse limitate

Introduciamo ora un limite nel numero di proiettili colorati che abbiamo a disposizione: come già fatto per le vite del giocatore, anche in questo caso utilizzeremo una variabile comune a tutti gli sprite che chiameremo “Serbatoio del colore”. Terremo la sua etichetta visibile, in basso a destra dello schermo (vedi immagine qui a fianco).

Aggiungiamo anche un nuovo sprite “arcobaleno” a partire dall’immagine “Rainbow” che troviamo nella libreria di Scratch: grazie allo script descritto qui sotto, lo sparpaglieremo per il campo per ottenere alcuni punti di rifornimento (anche questo sprite avrà quindi una variabile privata “distanza dall’aeroporto” che tiene traccia della sua posizione Y rispetto al punto in cui decolla l’aereo).



Tutti gli script di questo nuovo sprite (a parte quello che gestisce l’impatto tra King Kong e la goccia) sono molto simili a quelli dei gorilla, ed infatti ci conviene copiare quelli, trascinando qui gli script di King Kong; poi vanno modificati come descritto nell’immagine qui sotto: c’è il blocco iniziale che valorizza la variabile del serbatoio (❶) e crea le copie degli arcobaleni (❷); il blocco che si occupa dello spostamento (❸) è identico a quello dei gorilla; lo script che gestisce il contatto con l’aereo (❹) effettua il rifornimento del colore (❺), fa lampeggiare la scritta del serbatoio (❻) ed infine fa sparire l’arcobaleno esaurito (❼).

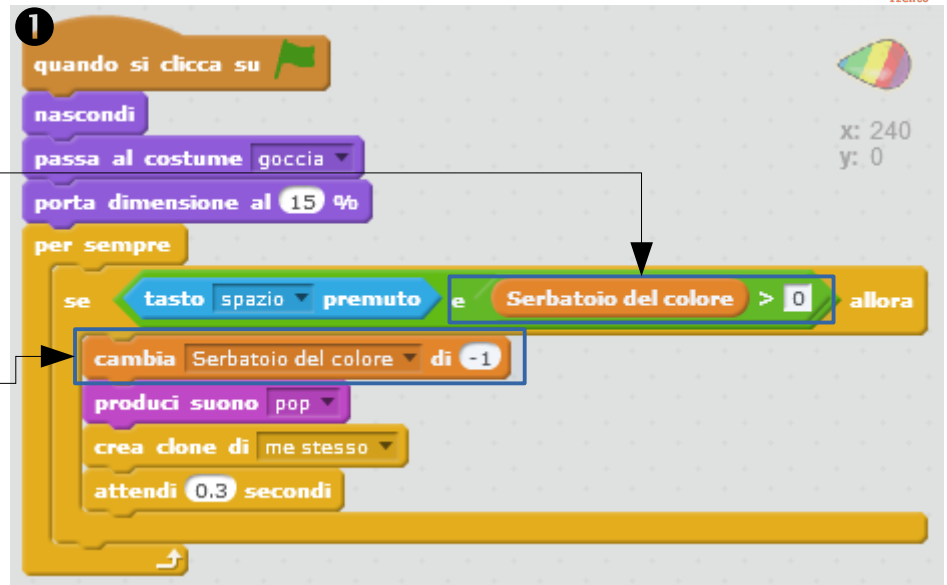


Se il serbatoio del colore può essere rifornito, chiaramente ci sarà anche il momento in cui questo viene svuotato: bisogna infatti apportare due piccole modifiche allo script della “goccia”, come indicato nell’immagine ①

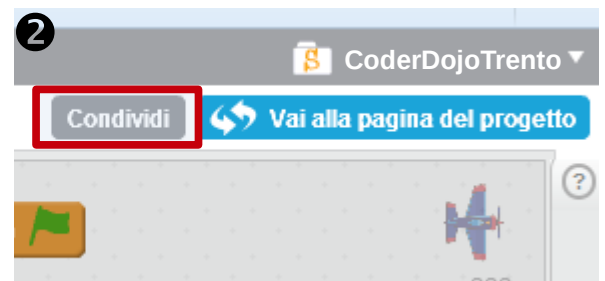


\* quando premiamo spazio, la goccia-proiettile parte solo se il serbatoio non è vuoto;

\* al momento della creazione di un nuovo clone della goccia attingiamo dal serbatoio: dobbiamo quindi abbassare di 1 il livello della nostra riserva di colore.



Ci siamo! Se vuoi mostrare il gioco ai tuoi amici, puoi farlo grazie al sito di [Scratch \(scratch.mit.edu\)](https://scratch.mit.edu): devi registrarti utilizzando un indirizzo e-mail valido (nella casella di posta troverai infatti una mail che ti permette di confermare l’iscrizione), caricare il progetto sul sito (crea un nuovo progetto e utilizza il comando “Carica dal tuo computer” del menu “File”), ed infine condividi il progetto con il link in alto a destra (vedi figura ②).



## 10. Migliorie varie

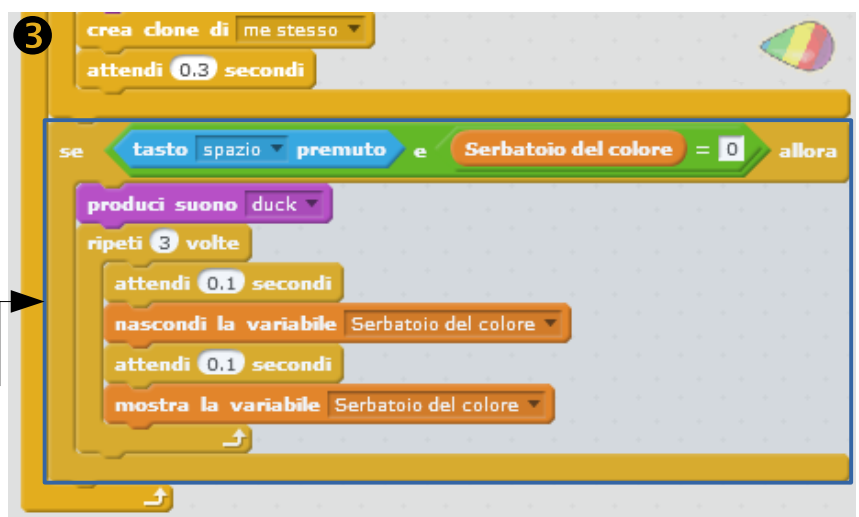
Hai salvato, testato, condiviso con i tuoi amici e ti sei divertito con il gioco costruito fino ad ora? Bene!

Ci sono comunque moltissime migliorie che possono essere introdotte per rendere il gioco ancora più interessante e divertente. Qui sotto vengono elencati un po’ di spunti, ma non c’è limite alla fantasia!

### Miglioria 1: avvertimento serbatoio del colore esaurito

Per avvertire il giocatore che il serbatoio del colore è vuoto, è possibile aggiungere questo blocco di codice dopo il “se” che fa partire la goccia (vedi immagine ③)

\* se il giocatore preme spazio ma in quel momento il serbatoio è vuoto, emettiamo un rumore di avvertimento e facciamo lampeggiare la scritta del serbatoio (come era stato fatto dagli script dello sprite “arcobaleno” durante il rifornimento).



## Miglioria 2: game over

Quando un gorilla colpisce l'aereo per l'ultima volta, il nostro gioco si limita ad eseguire un'istruzione "ferma tutto".

Possiamo curare un po' meglio la situazione ricorrendo ad una scritta "GAME OVER" da mettere a tutto schermo. Possiamo utilizzare lo strumento indicato nell'immagine **1** per creare un nuovo sprite disegnandolo noi stessi: appena attivato, ci troviamo davanti alla canvas vuota del costume del nuovo personaggio; usando il menù in basso a destra, passiamo in primo luogo a modalità vettoriale (vedi immagine **2**) per poter utilizzare lo strumento "Testo" (immagine **3**); clicchiamo sulla canvas e scriviamo appunto le parole "GAME OVER", quindi scegliamo il carattere che ci piace di più ed ingrandiamo per bene il testo, centrando la scritta all'interno del costume. Posizioniamo poi lo sprite all'interno della schermata del gioco.

Per far comparire la scritta al momento giusto, utilizzeremo i messaggi: King Kong, invece di usare "ferma tutto", notificherà a tutti gli sprite la fine del gioco mandando il messaggio "fine" (vedi **4**). Il nostro "game over" reagirà al messaggio, dal momento che il suo script sono quelli riportati in figura **5**.



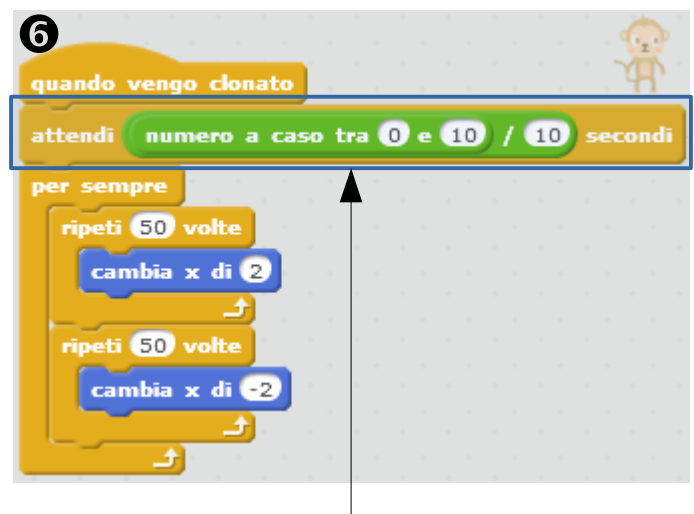
## Challenge 1: facciamo muovere i King Kong

Il gioco diventa molto più divertente se i gorilla possono muoversi, così sarà più difficile colpirli con le gocce di colore. Per introdurre lo spostamento sarà sufficiente aggiungere dei blocchi che si attivano quando lo sprite viene clonato, simili a quello descritto nell'immagine **6** qui a destra

In questo caso abbiamo un semplice spostamento a destra e sinistra, infatti lo script agisce solo sulla coordinata X.

Prova ad introdurre nuovi tipi di movimento, ricordandoti che se vuoi spostare King Kong verso l'alto o il basso, non devi aggiornare direttamente la coordinata Y, ma la variabile "distanza dall'aeroporto": sarà poi il blocco di codice che si occupa dello spostamento a mettere lo sprite nella giusta posizione.

Ricordati anche di introdurre un po' di casualità, facendo partire il movimento ad un istante casuale (come fatto nello script mostrato) o cambiando l'ampiezza degli spostamenti: se utilizzi lo stesso script per tutti i gorilla, c'è infatti il rischio di vedere tutti i movimenti sincronizzati come in un gigantesco balletto. :-D



## Challenge 2: gestiamo il livello di arrabbiatura dei King Kong



Come il giocatore dispone di un certo numero di vite da utilizzare per poter giocare prima che avvenga il GAME OVER, così possiamo introdurre un contatore per fare in modo che servano diverse gocce di colore per calmare del tutto i King Kong e quindi farli sparire dal campo di gioco.

Per gestire il “livello di arrabbiatura” utilizzeremo una variabile privata (selezioniamo cioè “Solo per questo sprite”, quando la creiamo) di King Kong, che porremo ad esempio a 3 al momento in cui lo sprite viene clonato; inoltre la decrementeremo di 1, ogni volta che una goccia di colore andrà a colpirlo e solo quando l’arrabbiatura sarà passata del tutto (ovvero quando la variabile varrà 0), faremo sparire il gorilla.

A differenza di quanto accade per la variabile “Vite” che viene sempre visualizzata in basso a sinistra durante tutto il gioco, non potremmo usare le etichette per mostrare il livello di arrabbiatura: ci sono diversi King Kong e non sarebbe facile capire se il gorilla che abbiamo di fronte è più o meno arrabbiato (e quindi se servono più gocce o una sola per farlo sparire). Possiamo però mostrare visivamente l’arrabbiatura, agendo ad esempio sulla velocità con cui vengono agitate le braccia, oppure il colore del gorilla o la sua dimensione (usa gli effetti che trovi sotto il menù “Aspetto”); se abbiamo un buon estro per il disegno, possiamo preparare tutta una serie di nuovi costumi (oppure importarli dalla libreria). Ecco alcuni esempi di arrabbiatura crescente a tre livelli. :-p



### Ringraziamenti

Questo tutorial:

- \* fa parte delle “risorse” del CoderDojo Trento,
- \* è stato scritto con il supporto di CoderDolomiti,
- \* ma soprattutto: è approvato da “Violet the pilot”!



CoderDojo  
Trento

Coder  
Dolomiti



Si ringraziano per l’ospitalità



Un grazie particolare a Steve Breen, per la concessione di utilizzo delle immagini tratte dal suo libro “Violet the pilot” (ISBN-13: 978-0803731257) pubblicato da Dial Books.



Le fotografie aeree sono state messe a disposizione dal Comune di Trento, che le distribuisce a titolo gratuito tramite il Servizio Sistema Informativo.



COMUNE DI TRENTO

